

Learning for Expertise Matching with Declination Prediction

Yujie Qian, Jie Tang

Department of Computer Science and Technology, Tsinghua University
qyj13@mails.tsinghua.edu.cn, jietang@tsinghua.edu.cn

Abstract. We study the problem of finding appropriate experts who are able to complete timely reviews and would not say “no” to the invitation. The problem is a central issue in many question-and-answer systems, but has received little research attention. Different from most existing studies that focus on expertise matching, we want to further predict the expert’s response: given a question, how can we find the expert who is able to provide a quality review and will agree to do it. We formalize the problem as a ranking problem. We first present an embedding-based question-to-expert distance metric for expertise matching and propose a ranking factor graph (RankFG) model to predict expert response. For online evaluation, we developed a Chrome Extension for reviewer recommendation and deployed it in the Google Chrome Web Store, and then collected the reviewers’ feedback. We also used the review bidding of a CS conference for evaluation. In the experiments, the proposed method demonstrates its superiority (+6.6-21.2% by MAP) over several state-of-the-art algorithms.

1 Introduction

Finding appropriate experts for given questions is a central task in the question-and-answer problem. For example, Quora and Zhihu¹ have attracted big-time experts in numerous fields to answer various questions. Another example is peer review, an important part of scientific publishing [21,12]. Many researchers consider peer review as part of their professional responsibility. Publishing groups also use the quality of peer reviews as an indicator of the success of a journal. At a high level, the problem is referred to as *expertise matching* — matching experts with questions. However, existing solutions for expertise matching are far from satisfactory. Peer review has long been criticized for being ineffective, slow and of low-quality. Statistics also show that about half of the questions on Quora only have one or do not have any answer². One challenge is how to find appropriate experts who are qualified to answer the corresponding questions. The other one is how to find experts who will agree to provide their answers. The latter is even more serious. One would expect that experts with sufficient knowledge would be the best to answer a given question. However, in practice, just to mention a few, American Political Science Review reported 2051 of 4516 reviewers accepted in 2013 (decline rate: 54.6%) [7] and a decline rate of 49.6% in 2014 [8]. Our preliminary statistics also show that 52.7% of the review invitations are declined or ignored.

¹ <http://quora.com>, <http://zhihu.com>

² <https://www.quora.com/What-percentage-of-questions-on-Quora-have-no-answers>

Fig. 1(a) illustrates the declination rate of review invitation from nine journals in our data. J1 and J2 (two materials science journals) suffer from a decline rate of almost 80%. Many top experts are inclined to say “no” for various reasons. As a result, finding appropriate experts who are able to complete timely reviews remains a challenge.

Quite a few studies have been conducted for automating the expert-question matching (details can be found in Section 5). However, most of this research focuses on the following setting: given a list of experts and a list of questions, how to find an optimal matching (with some constraints) between experts and questions.

In this paper, we study a more open question: given a new question, how to find appropriate experts who will *agree to answer* this question? The problem is much more challenging than the traditional matching problem, which is mainly concerned with the relevance between a question and an expert’s interest. However, factors that affect an expert to accept (or reject) a review invitation are more complicated, including not only the relevance of the question topic to the expert’s interest, but also the popularity of the question and whether the expert can gain attention from others. For peer review, other reasons for experts (researchers) to decline a review invitation include having too many reviews at hand and a tight deadline for completing the review [23,24]. Another survey of political science journals also finds that decline rate is related to the expert’s personal experience, e.g., top experts are more likely to decline than junior experts [2]. Technically, the challenge is how to design a principled approach to deal with the problem of expertise matching by considering the declination.

In this paper, we formalize the problem as a ranking problem and propose a ranking factor graph (RankFG) model to predict the willingness of experts to answer a given question. We introduce an embedding-based metric to measure the degree of expertise matching between the questions and experts, using them as features in the ranking model. RankFG is able to identify who is at a high risk of declining a review invitation. To empirically evaluate the proposed methodologies, we developed a Chrome Extension of reviewer recommendation and deployed it in the Google Chrome Web Store. About 30 journal editors downloaded the extension and used it for reviewer recommendations. Based on their feedback logs, the proposed method can clearly better predict the declination response than several state-of-the-art ranking algorithms. The proposed method is very general and can be flexibly applied to various scenarios. For example, we applied the proposed method to a major CS conference with around 1,000 submissions and 440 PCs to quantitatively evaluate the prediction quality. We use the bidding information as the ground-truth to evaluate whether an PC will be willing to review a submission or not. Experiments show that our method can significantly improve (+6.6-21.2%) the accuracy of expertise matching by considering declination.

2 Problem Definition

Given a question, our goal is to find experts with sufficient knowledge who are willing to review this paper.

We consider a social network $G = (V, E)$, where V is a set of $|V| = N$ experts and $E \subseteq V \times V$ is a set of relationships between experts. There can be various kinds of relationships in different social network. For example, in an academic social network,

the relationships include collaborations, same-affiliation, and same-nationality; while in a Quora-like network, the relationships include friendship, reply, and co-reply. Let \mathbf{a}_i denote expert v_i 's attributes, which could be one's interests, the questions she/he has replied, or simply the number of his/her posts. We use $A = \{\mathbf{a}_1, \dots, \mathbf{a}_N\}$ to denote the attributes of all experts. Given these definitions, we define our problem of expertise matching with declination as follows.

Problem 1. Expertise Matching with Declination Prediction. Let $G = (V, E, A)$ be an attribute-augmented social network. Given a particular question q , the goal is to design a predictive function such that we can suggest experts of high relevance to question q and of low risk of declining the review invitation, i.e.,

$$f : (G, q) \rightarrow Y \quad (1)$$

where $Y = \{y_1, \dots, y_N\}$ represents the prediction results for all experts in the network G and $y_i \in \{0, 1\}$ is a binary score indicating whether expert $v_i \in V$ has a high risk of declining the invitation.

The predictive function f takes the network G as input, which means that we should consider the network information in designing the predictive function. Additionally, another technical challenge is how to consider both expertise matching and the potential of declination.

3 The Approach

The straightforward method to deal with the expertise matching problem is to design a metric to quantify the similarity between question and expert, and then rank experts based on the similarity scores. However, the situation becomes different when taking declination into consideration. Besides, designing a *high-quality* similarity metric between question and expert is also a non-trivial task.

In this paper, we propose a ranking factor graph (RankFG) model. Specifically, given a question q , we first extract candidate experts through an information retrieval model. In order to quantify the similarity scores between questions and experts, we present an embedding-based matching algorithm. The algorithm leverages a constraint-based optimization to model the matching score between the two different types of entities – i.e., question and expert. Based on the similarity score, the RankFG model can learn and predict who has a high risk of declining the review.

3.1 Candidate Generation

Given a question q , we first use all words in the question to select a list of candidate experts.³ In particular, we use a language model to retrieve relevant experts from V .

³ In our implementation, we also tried to use a keyword extraction tool [29] to extract a number of keywords from the question and then use the keywords to select candidate experts.

Language model is one of the state-of-the-art approaches in information retrieval. It interprets the relevance between a document and a query word as a generative probability:

$$P(w|d) = \frac{N_d}{N_d + \lambda} \cdot \frac{N_d^w}{N_d} + (1 - \frac{N_d}{N_d + \lambda}) \cdot \frac{N_{\mathbf{D}}^w}{N_{\mathbf{D}}} \quad (2)$$

where N_d is the number of word tokens in document d , N_d^w is the word frequency (i.e., occurrence number) of word w in d , $N_{\mathbf{D}}^w$ is the number of word tokens in the entire collection, and $N_{\mathbf{D}}$ is the word frequency of word w in the collection \mathbf{D} ; λ is the Dirichlet smoothing factor and is commonly set according to the average document length in the collection [28]. Thus, the probability of the document model d generating a question q can be defined as $P(q|d) = \prod_{w \in q} P(w|d)$.

3.2 Expertise Matching

The language model only represents the matching between question and expert at a course-level. We present a more elaborate question-to-expert distance metric. The metric is inspired by [16], which was designed for estimating document distance.

Given all questions and all experts' interests, we first learn a numeric vector representation (also called word embedding) for each word. We use Word2vec with Skip-gram [18] to learn the word embeddings⁴. Word2vec is a shallow neural network architecture that consists of an input layer, a projection layer, and an output layer. The training objective is to use an input word to predict surrounding words in the context. Formally, we can have the following log-likelihood objective function,

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j}|w_t) \quad (3)$$

where T is the size of the training corpus and c is the window size around the center word w_t . The probability $p(w_{t+j}|w_t)$ is defined by a softmax function $p(w_O|w_I) = \frac{\exp(\mathbf{v}'_{w_O} \cdot \mathbf{v}_{w_I})}{\sum_{w=1}^W \exp(\mathbf{v}'_w \cdot \mathbf{v}_{w_I})}$, where \mathbf{v}_w and \mathbf{v}'_w are the input vector and output vector of word w , and W is the vocabulary size.

After the word embedding training, we obtain a m -dimensional embedding vector \mathbf{x}_i for each word $i \in \{1, \dots, n\}$. We consider both question and expert as a document. (For expert, we combine all the questions he/she answered or the papers he/she published.) Each document can be represented as a normalized bag-of-words (nBOW) vector $\mathbf{d} \in \mathbb{R}^n$, with each element $\mathbf{d}_i = N_d^i / (\sum_{j=1}^n N_d^j)$ standing for the weight of word i , where N_d^i is the word frequency of word i in the document. Finally, the question-to-expert distance $\text{QtoE}(q, v)$ can be defined by the following linear program,

$$\begin{aligned} \text{QtoE}(q, v) = \min_{\mathbf{T}_{ij} \geq 0} & \sum_{i=1}^n \sum_{j=1}^n \mathbf{T}_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|_2 \\ \text{subject to: } & \sum_{j=1}^n \mathbf{T}_{ij} = \mathbf{d}_i^q, \sum_{i=1}^n \mathbf{T}_{ij} = \mathbf{d}_j^v, \forall i, j \in \{1, \dots, n\}. \end{aligned} \quad (4)$$

⁴ We also tried CBOW, which results in an inferior performance comparing with Skip-gram.

where \mathbf{d}^q is the nBOW vector for question q and \mathbf{d}^v the vector of expert v . The linear program to find an optimal matching between the words in q and v , while each word i in q can be matched with different words j in v , with the weight \mathbf{T}_{ij} .

Complexity. The best time complexity for solving the above optimization problem is $O(p^3 \log p)$, where p is the number of unique words [20]. An approximate solution can result in a complexity of $O(p^2)$ [16], by removing each of the two constraints, and then combines the results.

3.3 Ranking Factor Graph

Previous works usually focused on expertise matching, but seldom considered whether the expert would decline the invitation. We propose a ranking factor graph (RankFG) model to predict how likely the expert is to accept or decline the invitation. The graphical model RankFG consists of two layers of variables: observations and latent variables. In our problem, each observation represent a question-expert pair $\{(q, v_i)\}$, and is associated with a latent variable y_i to represent whether the expert will agree or decline to answer the question (or review the paper). Local factor functions are defined to capture the relationships between an observation and its corresponding latent variable.

Local factor function: Captures the characteristics of each question-expert pair, including a relevance score between the paper and the expert, and any attributes associated with the expert. Defined as an exponential function

$$f(q, v_i, y_i) = \frac{1}{Z_a} \exp \left\{ \boldsymbol{\alpha}^T \boldsymbol{\psi}(q, v_i, y_i) \right\} \quad (5)$$

where $\boldsymbol{\psi}(\cdot)$ is the vector of feature functions defined between q and v_i with respect to the value of y_i ; $\boldsymbol{\alpha}$ is the weight of the features; and Z_a is a normalization factor.

Moreover, RankFG can leverage the data correlation to improve the prediction performance. For example, candidate experts from the same country (like USA) may say “no” at the same time to a question. From the paper reviewer data in our experiments, we have found three interesting correlations, same nationality, same affiliation, and collaboration. Fig. 1(b) and Fig. 1(c) shows statistics on the data in our experiments. We can see that the decline probability of a reviewer having a correlated expert who already declined is much higher than that of those without. On average, the probability that a reviewer will decline the invitation almost doubles if another reviewer of the same nationality has declined the invitation. In RankFG, such correlations can be defined as correlation factor functions among latent variables.

Correlation factor function: Captures the correlation between latent variables. Also defined as an exponential function

$$g(y_i, y_j) = \frac{1}{Z_b} \exp \left\{ \boldsymbol{\beta}^T \boldsymbol{\phi}(y_i, y_j) \right\} \quad (6)$$

where $\boldsymbol{\phi}(\cdot)$ is the vector of feature functions defined between y_i and y_j ; $\boldsymbol{\beta}$ is the weight of the features; and Z_b is a normalization factor.

By integrating the defined factor functions, we can define the following joint probability:

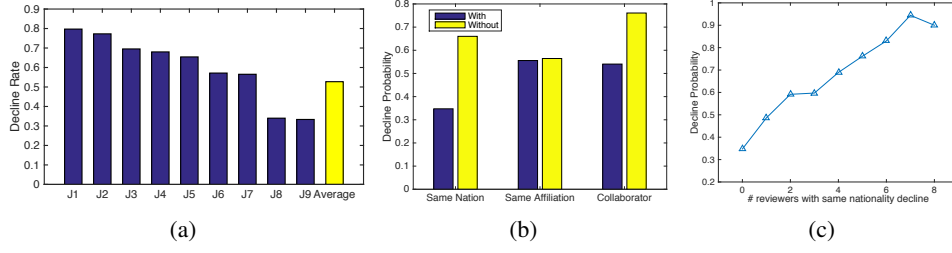


Fig. 1. (a) Decline rate of review invitations in our data, where J1-J9 represents different journals. (b) Decline probability of experts with or without correlated experts who have already declined. (c) Decline probability conditioned on the number of experts of same nationality who have already declined.

$$\begin{aligned}
P(Y|G) &= \prod_{v_i \in V} f(q, v_i, y_i) \prod_{(v_i, v_j) \in E} g(y_i, y_j) \\
&= \frac{1}{Z} \exp \left\{ \sum_{v_i \in V} \alpha^T \psi(q, v_i, y_i) + \sum_{(v_i, v_j) \in E} \beta^T \phi(y_i, y_j) \right\} \quad (7) \\
&= \frac{1}{Z} \exp \{ \theta^T \Phi \}
\end{aligned}$$

where $Z = Z_a Z_b$ is the normalization factor; $\theta = (\alpha^T, \beta^T)^T$ are parameters to estimate, and $\Phi = (\sum_{v_i \in V} \psi(q, v_i, y_i)^T, \sum_{(v_i, v_j) \in E} \phi(y_i, y_j)^T)^T$.

Combining Expertise Matching into RankFG Now we introduce how to combine the expertise matching scores into the proposed RankFG model. Specifically, given a question q and an expert v , we obtain a score $QtoE(q, v)$. We also defined other feature functions in the RankFG model. In principle, the feature functions can be instantiated in different ways to reflect our prior knowledge or intuitions for different applications. They can be defined as either binary or a real-valued. In our experiments, we define the local feature functions (ψ) which can be divided into the following three categories:

- **Basic statistics.** We define a set of statistics features for each potential expert. For example, in reviewer finding task, we use the features such as h -index, publication number, citation number, and the length of research experience.

- **Expertise matching.** We measure expertise matching between question and expert by $QtoE(q, v)$. We also consider other basic similarity scores as features, such as Jaccard similarity.

- **Organization.** We define a binary feature to indicate whether the reviewer comes from academia or industry.

Three correlation feature functions (ϕ) are also defined according to experts' relationships, which are:

- **Same-nationality.** This relationship exists when two experts come from the same country.

– **Same-affiliation.** This relationship exists when two experts come from the same affiliation.

– **Friendship.** This relationship exists when two experts have a friend relationship or have collaborated before.

For all three correlation factors, we use binary functions, i.e., $\phi_l(y_i, y_j) = 1$, if and only if such correlation exists.

Model learning Training the RankFG model involves finding a parameter configuration θ from a given training dataset, such that the log-likelihood objective function $L(\theta) = \log P(Y|G)$ can be maximized, i.e.,

$$\theta^* = \arg \max_{\theta} \log P(Y|G) \quad (8)$$

The optimization can be solved using gradient ascent algorithm. The gradient of each parameter θ wrt $L(\theta)$ is

$$\frac{\partial L(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta} (\theta^T \Phi - \log Z) = \Phi - \mathbb{E}_{P(Y|G)} \Phi \quad (9)$$

In this equation, the second term $\mathbb{E}_{P(Y|G)} \Phi$ is unmanageable to estimate because of the difficulty in determining the marginal probability $P(Y|G)$. There are several approximation methods. In our work, we choose Loop Belief Propagation (LBP) [27]. We first derive a factor graph from the original graph G , representing the factorization of the likelihood $P(Y|G)$. Then we apply the sum-product algorithm [15] to factor graph to compute the approximate marginal distributions.

Algorithm 1 describes the learning process of the RankFG model. During the training, we update the parameters iteratively until convergence. In each iteration, messages are transferred sequentially in a certain order. We randomly select a node as the root and perform breadth-first search on the factor graph to construct a tree. We update the messages from the leaves to the root, then from the root to the leaves. Based on the received messages from factors, we can calculate the marginal probabilities. Then we compute the gradient ∇ and update the parameters with learning step η .

Prediction Given the observed value \mathbf{x} and the learned parameters θ , the declination prediction is to find the most likely configuration of Y_q for a given question q . This can be obtained by:

$$Y_q = \arg \max_{Y_q} P(Y_q|G) \quad (10)$$

For inference, we use the max-sum algorithm to find the values of Y_q that maximize the likelihood. This max-sum algorithm is similar to the sum-product algorithm, except for calculating the message according to *max* instead of *sum* in message passing functions.

4 Experiments

To empirically evaluate the proposed methodologies, we conduct the experiments in the peer review task, which is finding reviewers for academic papers.

Algorithm 1 Learning algorithm for RankFG.**Input:** Query questions $Q = \{q\}$, $G = (V, E, A)$, and the learning rate η ;**Output:** learned parameters θ ;

```

 $\theta \leftarrow 0$ ;
repeat
  for  $q \in Q$  do
     $L \leftarrow$  initialization list;
    Factor graph  $FG \leftarrow \text{BuildFactorGraph}(L)$ ;
    repeat
      for  $v_i \in L$  do
        Update the messages of  $v_i$  according to sum-product update rule [15];
      end for
    until all messages  $\mu$  do not change;
    for  $\theta_i \in \theta$  do
      Calculate gradient  $\nabla_i$  according to Eq. 9;
      Update  $\theta_i^{new} = \theta_i^{old} + \eta \cdot \nabla_i$ ;
    end for
  end for
until converge;

```

4.1 Experimental Setup

Datasets We evaluate our method in the following three different datasets:

Relevance: this dataset includes 86 papers and 2,048 candidate reviewers. We asked the PhD students from the author’s lab to make relevance judgments. We simply considered the relevance as binary: relevance and irrelevance. In this dataset, we focus on evaluating how the method can find *relevant and qualified* experts to review the paper.

Response: this dataset was collected from our Chrome extension for reviewer recommendation. It includes 183 papers submitted to nine journals and 827 reviewers’ responses. Among the responses, 391 are “agree”, while the rest are viewed as “decline” (including “unavailable” and “no response”). Figure 1(a) shows the declination rate. In Response, we focus on evaluating how our method can find experts who are *willing to review*, and reduce reviewer declination rate.

Conference: this dataset comes from the reviewer bidding of a major CS conference. It contains 935 submitted papers and 440 Program Committee members. PC members each choose several papers they would like to review. We consider their choices as “agree”, and randomly sample the same number of PCs from the others as “decline”. In Conference, we aim to apply the model in a *conference reviewer assignment* task.

In each dataset, we randomly select 60% as training data and consider the remaining 40% as test data. We perform each experiment 10 times (including partition of the training and test data) and report the average score.

Comparison Methods We compare the following methods in the experiment:

Jaccard Similarity: We calculate Jaccard similarity coefficient between the paper’s keywords and the reviewer’s research interests. The recommendation is then made based on the rank of similarity score.

Table 1. Results of reviewer recommendation in three datasets (%).

Method	Relevance				Response				Conference			
	P@5	P@10	MAP	Rprec	P@3	P@5	MAP	Rprec	P@5	P@10	MAP	Rprec
Jaccard	64.2	57.8	67.8	59.8	38.9	40.3	58.5	36.7	57.3	52.8	70.3	58.0
WMD	68.6	59.1	70.4	63.5	39.5	40.0	57.0	36.6	58.0	53.1	71.0	59.1
SVM-Rank	64.4	58.9	71.3	65.0	43.6	42.3	64.8	44.7	61.2	54.2	74.4	62.6
RankFG	64.5	58.5	70.8	64.3	47.5	44.0	69.1	51.8	61.9	54.6	75.6	64.4

Word Mover’s Distance (WMD): We apply Word Mover’s Distance [16] as the measure of expertise matching, and then rank the candidates by WMD.

SVM-Rank: We consider a learning to rank approach that uses the same training data as RankFG. Specifically, we use the implementation of SVM-Rank [9].

RankFG: The proposed method, which trains a RankFG model to make reviewer recommendations.

Evaluation Measures We consider the problem a ranking problem and evaluate different methods with the following metrics: Precision for the top-N results (P@N), Mean Average Precision (MAP), and R-prec. In the evaluations, we only consider the experts collected in our datasets.

All the experiment codes are implemented in C++ and Python, and the evaluations are performed on an x86-64 machine with 2.70GHz Intel Core i5 CPU and 8GB RAM.

4.2 Experiment Results

Performance Analysis We compare the performance of all methods on three datasets. Table 1 lists the performance comparison in three datasets. We find in Relevance, WMD significantly outperforms Jaccard and has the best performance of P@N within four methods. SVM-Rank and RankFG have lower P@N because the relevance between paper and reviewer mostly depends on content similarity while not other features and correlations. On the other hand, in Response, the proposed RankFG achieves the best result. RankFG leverages the reviewer correlation and thus improves the performance. It also confirms the discovery in the survey [23] that whether a reviewer agrees to review depends not only on expertise matching, but also on other factors. In the Conference dataset, RankFG still outperforms other methods. It suggests that the RankFG can also be applied in conference paper-reviewer assigning.

Factor Contribution Analysis We now analyze how different factors can help find reviewers. We examine the contribution of different features by removing each of them. Fig. 2(a) shows the result, using MAP score as the evaluation metric. We can see in Relevance and Conference, the performance mainly depends on expertise matching. Removing statistical or organization features does not significantly affect the performance. In Response, the situation is different. The MAP score drops when ignoring each kind of feature, especially statistics. It confirms the finding in a recent survey [2],

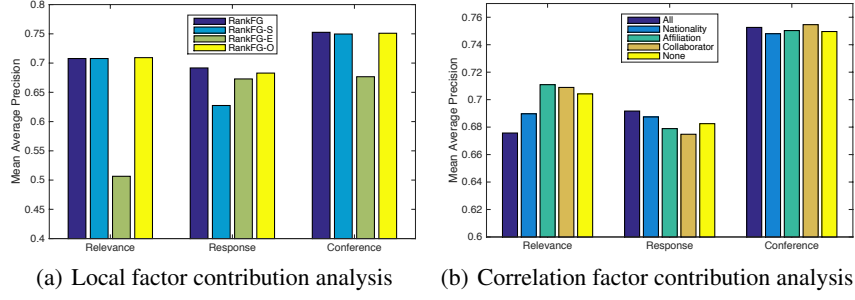


Fig. 2. (a) **Local factor contribution analysis:** RankFG-S, RankFG-E and RankFG-O each stands for ignoring statistics, expertise matching, or organization; (b) **Correlation factor contribution analysis:** Nationality, Affiliation, and Collaborator each stands for using only same-nationality, same-affiliation, or collaborator edges; None stands for removing all correlations.

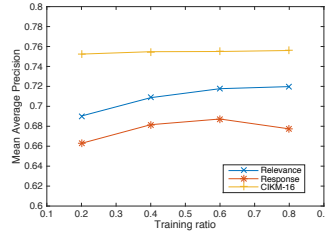


Fig. 3. Training/test ratio analysis of RankFG model.

which says “decline” is related to the reviewer’s personal academic status. Senior researchers tend to decline invitations more often.

We also analyze how correlation contributes to the model. Fig. 2(b) shows the analysis result. From the figure, we find that the contribution of correlation varies in different circumstances. Same-affiliation and collaborator have a positive effect in Relevance and Conference, but a negative effect in Response. However, the combination of these two correlations with same-nationality helps improve the prediction performance in Response. Same-nationality seems only work in Response among three datasets. It is reasonable since nationality is not related to someone’s research interests or review ability. So we remove same-nationality from other experiments on Relevance and Conference.

Training/Test Ratio Analysis We provide further analysis on the effect of the training ratio on our RankFG model. Fig. 3 shows the experiment results when varying the percentage of training data. In general, we can see a rising trend as the percentage of training data increases. In Conference dataset, the increase is relatively subtle since this dataset is the largest among the three. This result indicates the positive effect of training data size on the recommendation performance of our model. We also conduct an experiment to analyze the convergent property of the RankFG model. The learning

algorithm shows a very good convergent property. Roughly speaking, it takes 200-2000 iterations to converge on the three different datasets.

5 Related Work

The research of expertise matching can be traced back to 20 years ago. Existing methods mainly fall into two categories: probabilistic models and optimization models.

Probabilistic models try to improve the expertise matching accuracy between experts and papers. Early works such as latent semantic indexing [3] and keyword matching [5] tried to solve the problem in an information retrieval system. Later, Mimno *et al.* compared several language models and proposed Author-Persona-Topic model [19]. Karimzadehgan *et al.* proposed reviewer aspect modeling and paper aspect modeling, and tried to model multiple aspects of expertise [11].

Optimization model concentrates on solving the optimization problem of constructing panels between a list of reviewers and a list of papers. These models usually consider the constraints in conference paper-reviewer assignment tasks [1], such as conflict of interests, paper demand and reviewer workload. Many methods have been proposed, such as search and greedy algorithm [17,13], integer linear programming [10], network flow [4] and minimum cost flow [22].

Recently, a few systems have also been developed to make reviewer recommendations [26], help with conference reviewer assignment [14] or identifying reviewers for proposals [6]. Wu *et al.* [25] presents a patent partner recommendation framework in enterprise social networks, which also uses a ranking factor graph model.

In this paper, our general goal is to recommend appropriate experts and predict the willingness of them to accept the invitation.

6 Conclusions

In this paper, we study predicting experts response in expertise matching. Our goal is to find experts not only relevant to a question/paper, but also will not decline the invitation. We present an embedding-based question-to-expert distance metric, and propose a ranking factor graph (RankFG) model to find the proper experts. RankFG leverages both expertise matching and other attributes and correlation between experts. To fairly evaluate the proposed methodologies, we conduct experiments on three datasets: Relevance, Response, and Conference. Comparing with several state-of-the-art methods, our method can significantly improve the performance of expertise matching.

References

1. Benferhat, S., Lang, J.: Conference paper assignment. *Int. J. Intell. Syst.* 16(10), 1183–1192 (2001)
2. Djupe, P.A.: Peer reviewing in political science: New survey results. *PS: Political Science & Politics* 48(02), 346–352 (2015)
3. Dumais, S.T., Nielsen, J.: Automating the assignment of submitted manuscripts to reviewers. In: *SIGIR'92*. pp. 233–244 (1992)

4. Goldsmith, J., Sloan, R.H.: The ai conference paper assignment problem. In: AAAI'07. pp. 53–57 (2007)
5. Haym, C.B., Hirsh, H., Cohen, W.W., Nevill-manning, C.: Recommending papers by mining the web. In: IJCAI'99. pp. 1–11 (1999)
6. Hettich, S., Pazzani, M.J.: Mining for proposal reviewers: lessons learned at the national science foundation. In: KDD'06. pp. 862–871 (2006)
7. Ishiyama, J.: Annual report of the editors of the american political science review, 2012–2013. PS: Political Science & Politics 47(02), 542–545 (2014)
8. Ishiyama, J.: Report of the editors of the american political science review, 2013–2014. PS: Political Science & Politics 48(02), 396–399 (2015)
9. Joachims, T.: Training linear svms in linear time. In: KDD'06. pp. 217–226. ACM (2006)
10. Karimzadehgan, M., Zhai, C.: Constrained multi-aspect expertise matching for committee review assignment. In: CIKM'09. pp. 1697–1700 (2009)
11. Karimzadehgan, M., Zhai, C., Belford, G.: Multi-aspect expertise matching for review assignment. In: CIKM'08. pp. 1113–1122 (2008)
12. Kassirer, J.P., Campion, E.W.: Peer review: crude and understudied, but indispensable. JAMA 272(2), 96–97 (1994)
13. Kou, N.M., Hou, U.L., Mamoulis, N., Gong, Z.: Weighted coverage based reviewer assignment. In: SIGMOD'15. pp. 2031–2046. ACM (2015)
14. Kou, N.M., Mamoulis, N., Li, Y., Li, Y., Gong, Z., et al.: A topic-based reviewer assignment system. In: VLDB'15. pp. 1852–1855 (2015)
15. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory 47(2), 498–519 (2001)
16. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From word embeddings to document distances. In: ICML'15 (2015)
17. Long, C., Wong, R.C.W., Peng, Y., Ye, L.: On good and fair paper-reviewer assignment. In: ICDM'13. pp. 1145–1150. IEEE (2013)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
19. Mimno, D., McCallum, A.: Expertise modeling for matching papers with reviewers. In: KDD'07. pp. 500–509 (2007)
20. Pele, O., Werman, M.: Fast and robust earth mover's distances. In: ICCV'09. pp. 460–467. IEEE (2009)
21. Rennie, D.: Editorial peer review: its development and rationale. Peer review in health sciences pp. 3–13 (1999)
22. Tang, W., Tang, J., Lei, T., Tan, C., Gao, B., Li, T.: On optimization of expertise matching with various constraints. Neurocomputing 76(1), 71–83 (2012)
23. Tite, L., Schroter, S.: Why do peer reviewers decline to review? a survey. Journal of epidemiology and community health 61(1), 9–12 (2007)
24. Willis, M.: Why do peer reviewers decline to review manuscripts? a study of reviewer invitation responses. Learned Publishing 29(1), 5–7 (2016)
25. Wu, S., Sun, J., Tang, J.: Patent partner recommendation in enterprise social networks. In: WSDM'13. pp. 43–52 (2013)
26. Yang, K.H., Kuo, T.L., Lee, H.M., Ho, J.M.: A reviewer recommendation system based on collaborative intelligence. In: WI-IAT'09. vol. 1, pp. 564–567. IET (2009)
27. Yedidia, J.S., Freeman, W.T., Weiss, Y., et al.: Generalized belief propagation. In: NIPS'00. vol. 13, pp. 689–695 (2000)
28. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: SIGIR'01. pp. 334–342 (2001)
29. Zhang, K., Xu, H., Tang, J., Li, J.: Keyword extraction using support vector machine. In: WAIM'06. pp. 85–96 (2006)